TALEDGE ชั้นวางเรื่องราว

โดย

นาย สิรวิชญ์ ทิมสุวรรณ

นางสาว ณิชาภัทร ชมภูน้อย

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

รายงานโครงงาน

โครงงาน: TALEDGE: ชั้นวางเรื่องราว

กลุ่ม: CS369 กลุ่ม 11

สมาชิกในกลุ่ม:

1. 6309682067 นาย สิรวิชญ์ ทิมสุวรรณ (กาย)
2. 6309610134 นางสาว ณิชาภัทร ชมภูน้อย (นุ๊ก)

i. บทนำ

ความเป็นมาและความสำคัญของโครงงาน

ในยุคปัจจุบัน การเขียนนิยายได้รับความนิยมเพิ่มขึ้นอย่างต่อเนื่อง ทั้งในรูปแบบของสิ่งพิมพ์และบนแพลตฟอร์มออนไลน์ อย่างไรก็ตาม นักเขียนมักประสบปัญหาเกี่ยวกับการจัดการโครงเรื่อง ตัวละคร และเหตุการณ์ต่าง ๆ โดยเฉพาะเมื่อเนื้อหามีความซับซ้อนมากขึ้น ทำให้การวางพล็อตและจัดการองค์ประกอบต่าง ๆ กลายเป็นเรื่องที่ใช้เวลาและเกิดข้อผิดพลาดได้ง่าย

โครงงาน “TALEDGE: ชั้นวางเรื่องราว” จึงถูกพัฒนาขึ้นในรูปแบบของเว็บแอปพลิเคชัน เพื่อช่วยให้นักเขียนสามารถวางแผนโครงเรื่อง จัดการตัวละคร และพัฒนาเนื้อหาได้อย่างเป็นระบบ เพิ่มความสะดวกในการจัดการองค์ประกอบของนิยาย และลดอุปสรรคในกระบวนการสร้างสรรค์งานเขียน

วัตถุประสงค์

1. เพื่อพัฒนาเว็บแอปพลิเคชันสำหรับช่วยในการวางพล็อตนิยาย
2. เพื่อจัดการข้อมูลตัวละคร เหตุการณ์ และองค์ประกอบอื่นๆ ได้อย่างเป็นระบบ
3. เพื่อเพิ่มประสิทธิภาพและลดความซับซ้อนในกระบวนการเขียนนิยาย

ii. การออกแบบระบบ

Architecture Overview

ระบบ TALEDGE ออกแบบในลักษณะ Client-Server โดยมีการแยกส่วน Frontend และ Backend อย่างชัดเจน โดยมีการใช้ RESTful API เป็นตัวกลางในการเชื่อมต่อข้อมูลระหว่างสองฝั่ง ระบบสามารถสรุปได้ดังนี้:

- **Client (Frontend):** React.js
- **Server (Backend):** Node.js + Express.js
- **Database:** MongoDB Atlas
- **Hosting:** Netlify (Frontend) และ Render (Backend)

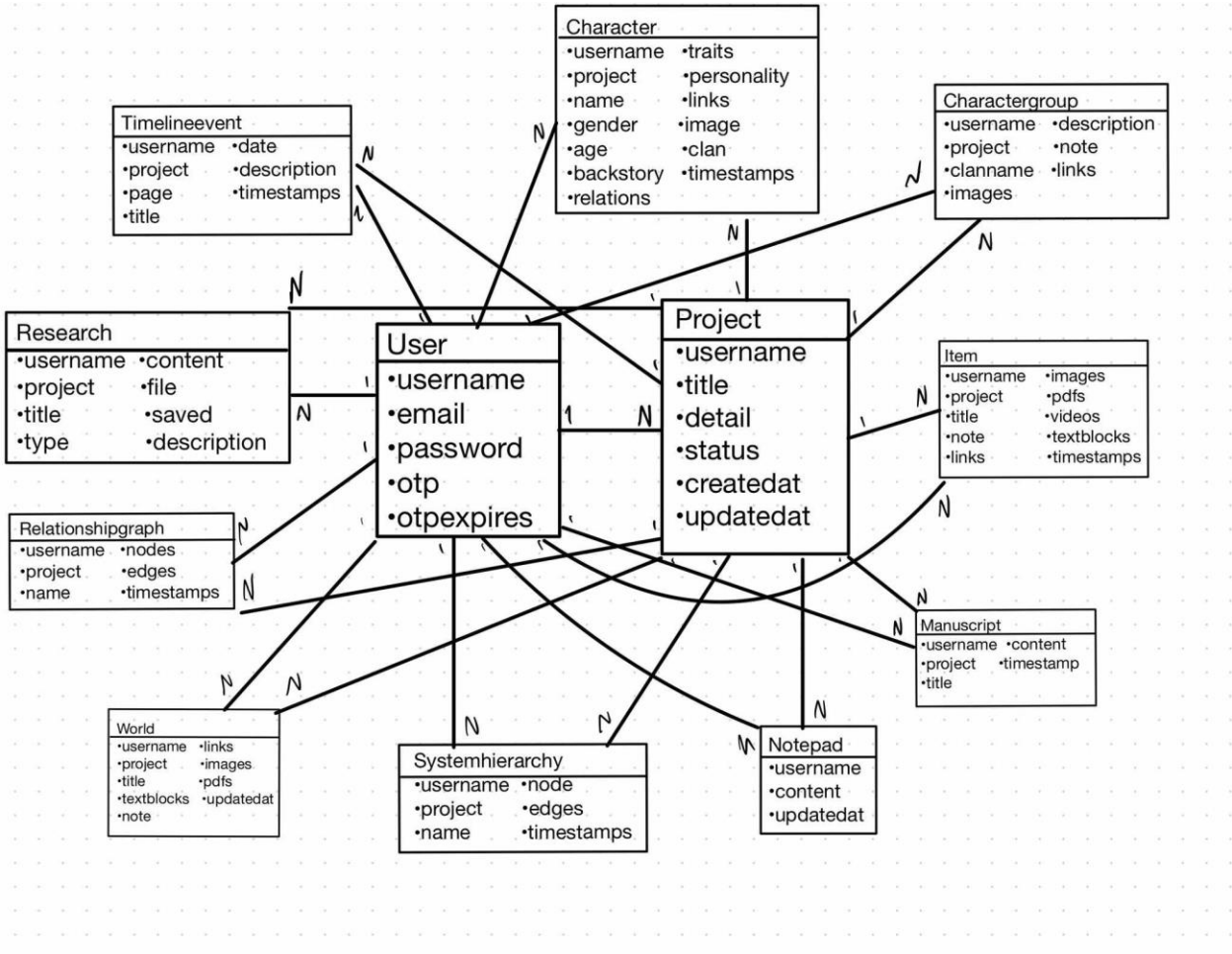
เทคโนโลยีที่ใช้

หมวดหมู่	เทคโนโลยี
Frontend	HTML5, CSS3, JavaScript (ES6), React.js
Backend	Node.js, Express.js, Python (Flask)
Database	MongoDB Atlas (NoSQL)
API	RESTful API, Flask API (สำหรับพจนานุกรม)
Tools	Git, VS Code, Postman

การออกแบบฐานข้อมูล

ระบบใช้ฐานข้อมูล NoSQL โดยออกแบบเป็น Document-Based Structure มี Schema หลักๆ ได้แก่:

- User: ข้อมูลผู้ใช้งาน เช่น username, email, password
- Project: ชื่อโปรเจกต์และ metadata
- Character, Event, Item, World, Research, System: องค์ประกอบที่เกี่ยวข้องกับโครงเรื่อง

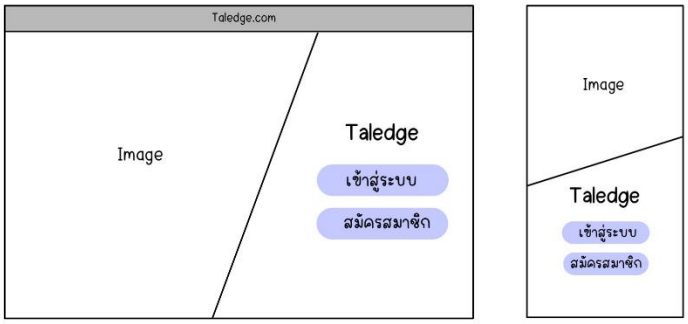


การออกแบบส่วนต่อประสาน (UI/UX)

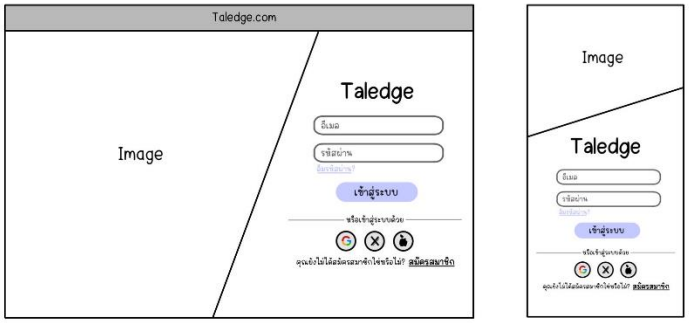
อินเทอร์เฟซของระบบมี Sidebar สำหรับเลือก Section ต่าง ๆ เช่น Timeline, Characters, Manuscript เป็นต้น

ภาพ Mockup ประกอบ:

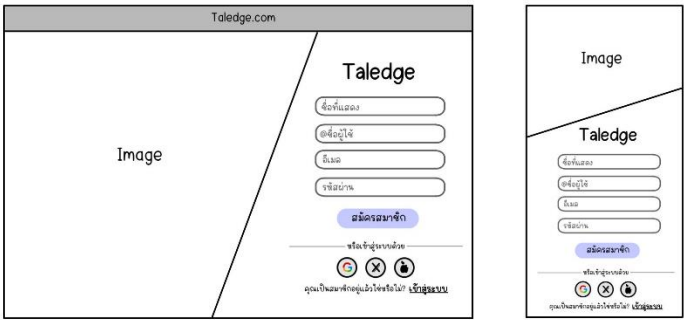
หน้าเริ่มต้น



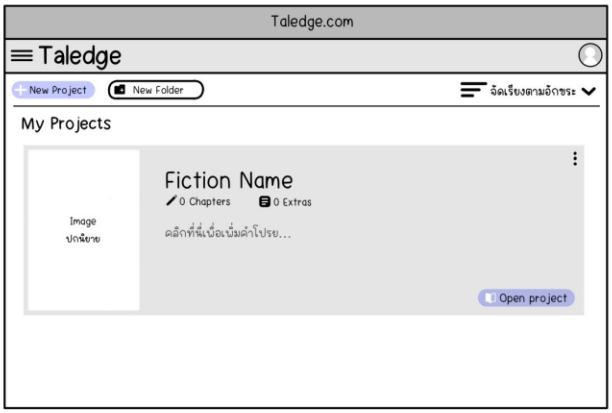
หน้าเข้าสู่ระบบ



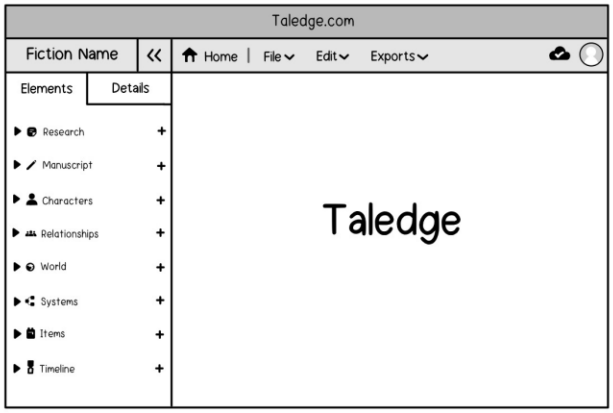
หน้าสมัครสมาชิก



หน้า Home/แดชบอร์ดส่วนตัว

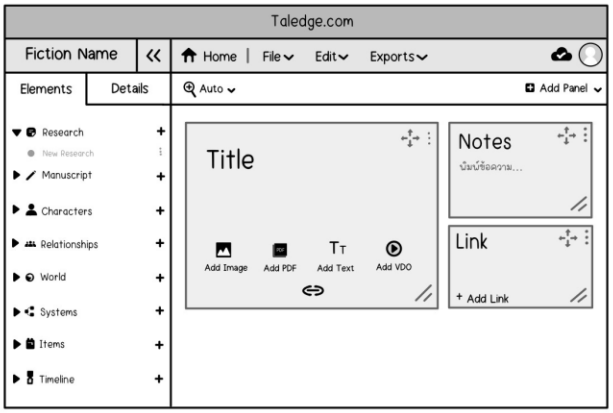


หน้าเริ่มต้น Project

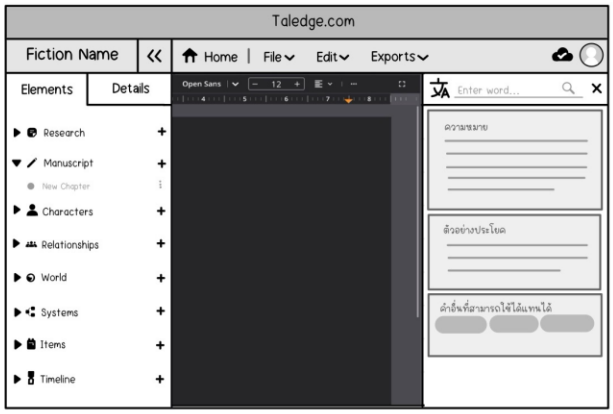


หน้า Research

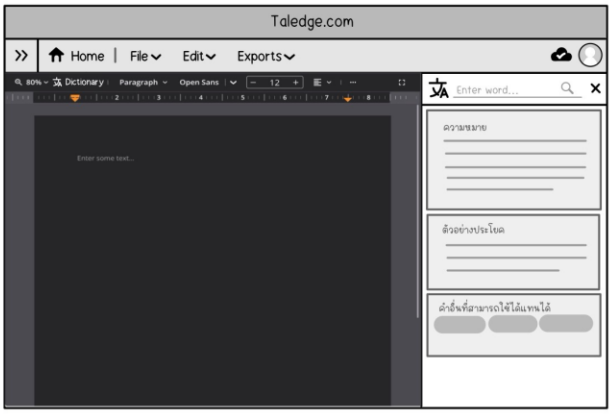
สำหรับเก็บไอเดียต่างๆ ข้อมูลที่เอามาใช้ในนิยาย



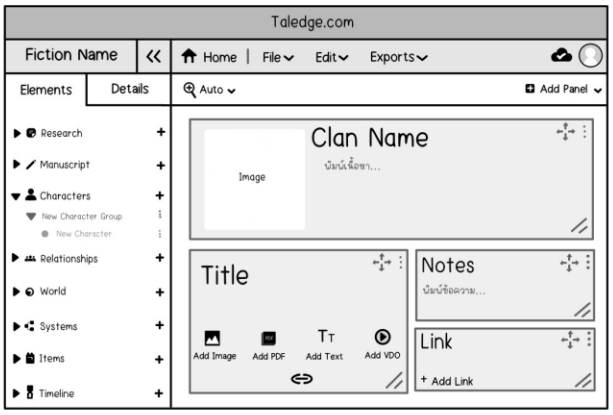
หน้าพิมพ์นิยาย



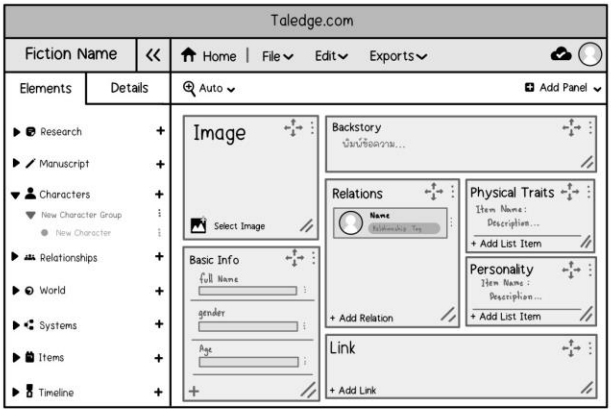
พับจอ elements



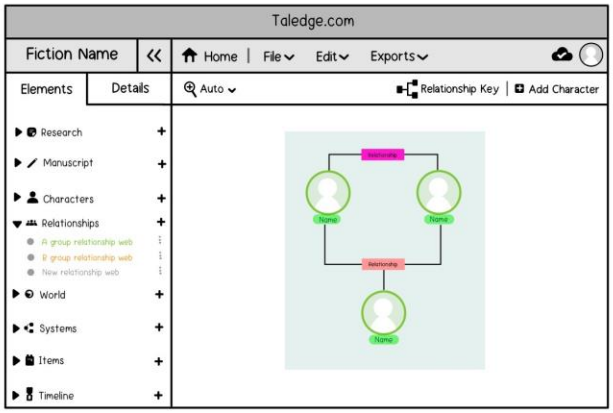
หน้า Characters Group Character



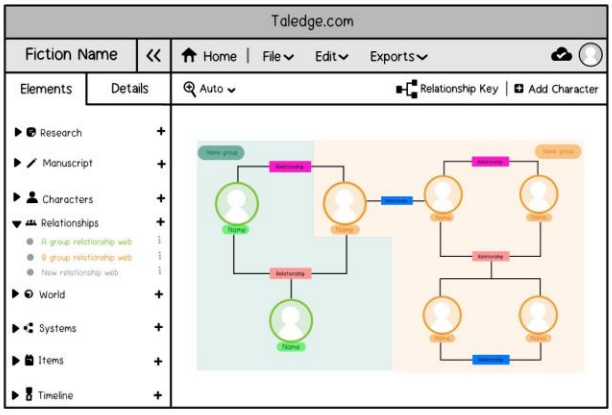
หน้า Characters New Character



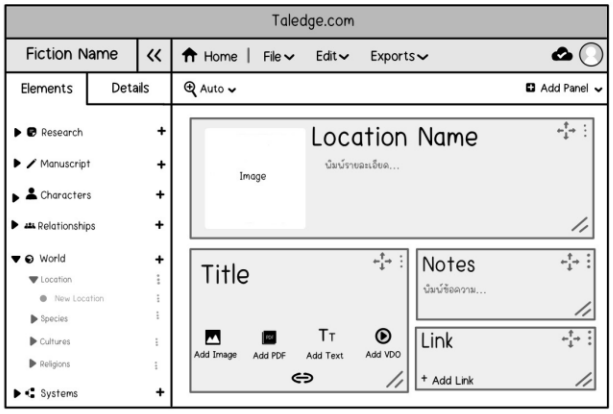
หน้า Relationship A group relationship web



หน้า Relationship New relationship web ความสัมพันธ์ระหว่างสอง group

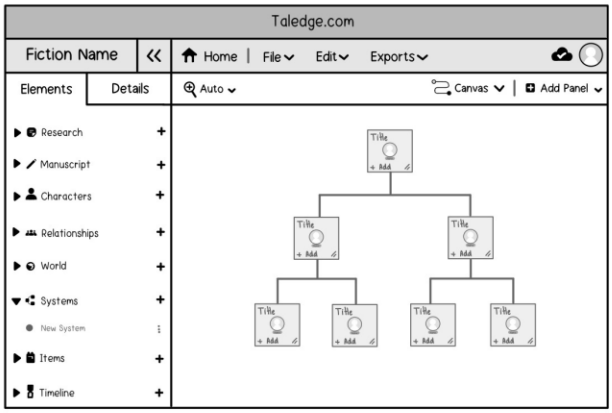


หน้า World building

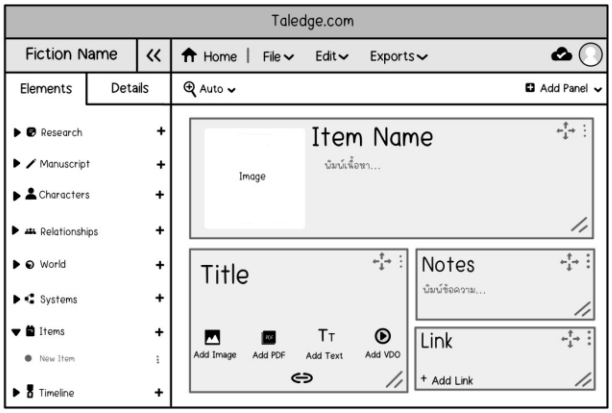


หน้า Systems

ตัวอย่าง class system

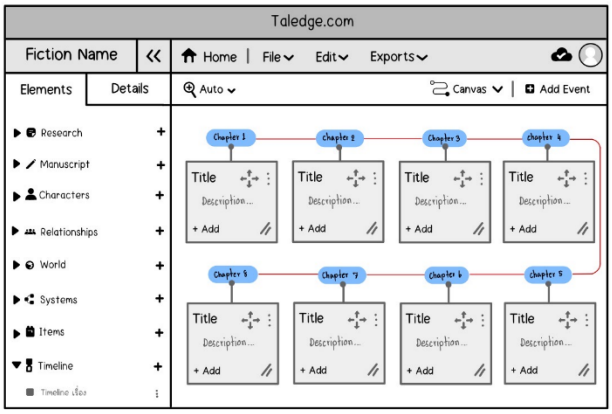


หน้า Item

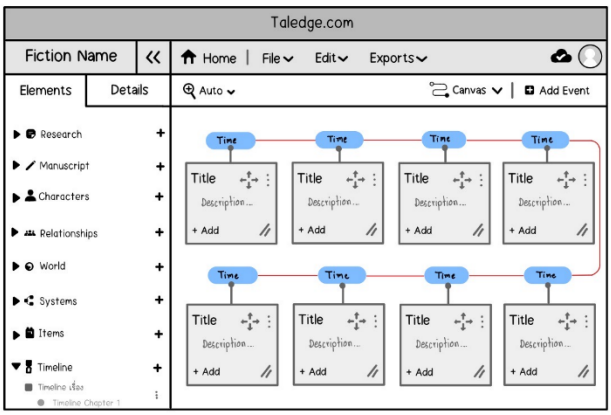


หน้า Timeline

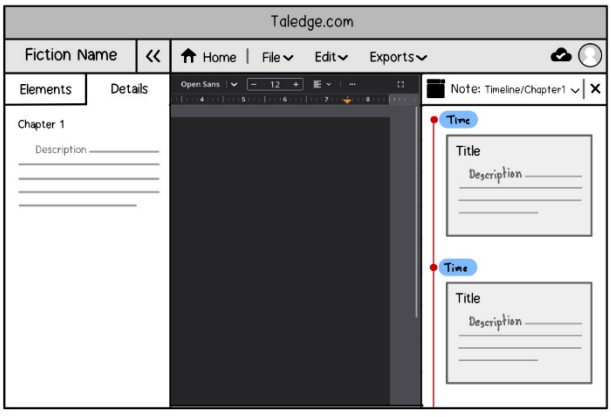
ตาม timeline เรื่อง



ตาม timeline chapter



เรื่องไปหน้าเขียนนิยาย



iii. การพัฒนา

Frontend

พัฒนาโดยใช้ React.js มีการใช้ Hooks (useState, useEffect) สำหรับจัดการ State และ Lifecycle Component ออกแบบเป็น Component-Based เพื่อแยกส่วนการทำงานได้ชัดเจน เช่น: TimelinePanel CharactersPanel ManuscriptPanel เป็นต้น

Backend

พัฒนาโดยใช้ Node.js และ Express.js สร้าง API Endpoint สำหรับการจัดการข้อมูล CRUD ของโปรเจกต์ ตัวละคร เหตุการณ์ ฯลฯ โดยมีการเชื่อม MongoDB ด้วย mongoose

Python API (app.py)

ในระบบยังมีการพัฒนา API เพิ่มเติมด้วยภาษา Python โดยใช้ Flask ซึ่งใช้สำหรับฟังก์ชันเสริม เช่น การประมวลผลภาษาธรรมชาติ (NLP), พจนานุกรม หรือบริการแปลภาษาเบื้องต้น โดย app.py จะรับคำศัพท์จากผู้ใช้ ส่งไปยัง PyThaiNLP และส่งผลลัพธ์กลับในรูปแบบ JSON

ไลบรารีที่ใช้ใน Python API:

- Flask: สำหรับสร้าง Web API
- PyThaiNLP: สำหรับตัดคำและเรียกความหมายภาษาไทย
- nltk: สำหรับเชื่อมฐานข้อมูล WordNet
- flask-cors: สำหรับรองรับการเรียก API จาก React frontend

เทคนิคที่ใช้

- ระบบลืมรหัสผ่าน (Forgot Password) ถูกพัฒนาโดยใช้โทเคนเฉพาะ (Token) ซึ่งส่งไปยังอีเมลของผู้ใช้ผ่านบริการ Nodemailer โดยลิงก์จะเชื่อมไปยังหน้า Reset Password และทำการยืนยัน token กับรหัสผ่านใหม่บนฝั่ง Backend อย่างปลอดภัย
- JWT สำหรับจัดการ Token Authentication
- React Router DOM สำหรับการเปลี่ยนหน้า
- Drag-and-Drop ในการเชื่อมโยงความสัมพันธ์ของตัวละคร

โครงสร้างของ Source Code

พวกออกแบบมาเป็นระบบจัดการงานเขียนนิยาย โดยแบ่งโมดูลตามหน้าที่ พร้อมการเชื่อมต่อฐานข้อมูล MongoDB โดยแยกข้อมูลตาม username และ project เพื่อความปลอดภัยและการทำงานหลายผู้ใช้ได้พร้อมกัน

โครงสร้างหลักของระบบ

ส่วนแรกคือ Frontend (React) โดยเราจะใช้ React + React Router สำหรับหน้าเว็บเพื่อทำให้ไม่ต้องโหลดหน้าใหม่ทุกครั้งที่สลับเมนู และมี State management ในตัว ทำให้จัดการข้อมูลในแต่ละหน้าสะดวก โดย React Router จะทำหน้าที่สร้าง navigation และ URL routing ได้ง่าย และ รองรับการส่ง state ไปหน้าอื่น

ต่อมาเราใช้ useState, useEffect เพื่อดึงข้อมูล, fetch() api จาก backend โดยใช้ useState สำหรับจัดการสถานะภายในแต่ละ componentและใช้ useEffect ดึงข้อมูลจาก backend เมื่อ component ถูก mount เช่น ดึงตัวละครจาก MongoDB ทันทีเมื่อเข้า หน้าของ CharacterList และเราก็ทำการเก็บข้อมูลของ user ใน localStorage เช่น username, token, project เพื่อที่ทำให้ไม่ต้องส่งข้อมูลซ้ำทุกครั้ง ง่ายต่อการตรวจสอบสิทธิ์ในการให้ user เข้าใช้งาน

และเรามีการใช้ CSS ในการตกแต่ง UX/UI ของระบบแบบแยกไฟล์ของแต่ละส่วนเพื่อที่จะทำให้ง่ายต่อการจัดการจะได้รู้ว่ากำลังทำในของส่วนไหนอยู่

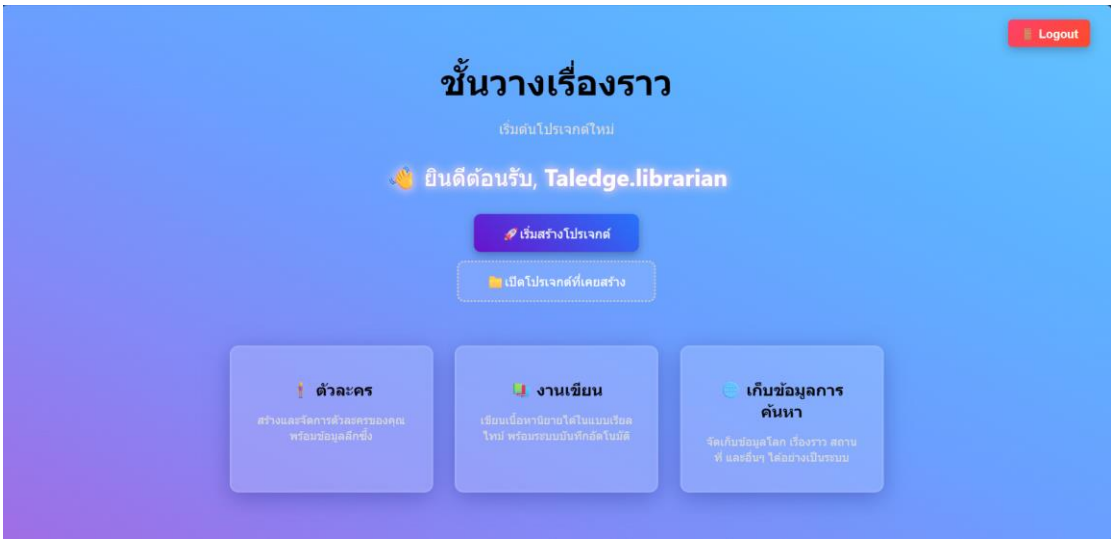
ในส่วนของ Backend พวก Express Rest ต่างๆ เราจะทำการ REST API ตาม Fiction ต่างๆ ของเราไม่ว่าจะเป็น characters, projects, system, research และอื่นๆ จากนั้น เชื่อมต่อ MongoDB ด้วย Mongoose โดยที่ข้อมูลทุกชุดถูกผูกอยู่กับ username และ project ก็คือ เราจะเก็บข้อมูลไว้กับ username นั้นๆ และแยกข้อมูลตามแต่ละ ProjectID ของ Username นั้นๆ เพื่อที่จะไม่ทำให้ข้อมูลทุกชุดถูกเก็บอยู่ใน username เดียว เพราะถ้าอย่างนั้นแล้วจะทำให้เกิดปัญหาตอนที่ username สร้าง Project ใหม่ขึ้นมาอีก Project หนึ่ง จะทำให้ข้อมูลของอีก Project หนึ่งขึ้นมาแสดงผลรวมกับข้อมูลของอีก Project หนึ่ง ซึ่งนั้นผิดต่อวัตถุประสงค์ของเราโดยที่เราต้องการที่จะรองรับการทำงานหลายๆ Project ของ user เดียวกัน อีกทั้งการแยกเก็บข้อมูลตาม username ก็เพื่อเป็นการรองรับการทำงานของหลาย user

iv. การทดสอบและการแสดงหน้าจอผลลัพธ์

ระบบได้ทำการทดสอบในระดับ Function Testing และ Manual UI Testing พบว่าฟังก์ชันทำงานได้ตรงตามที่ออกแบบ

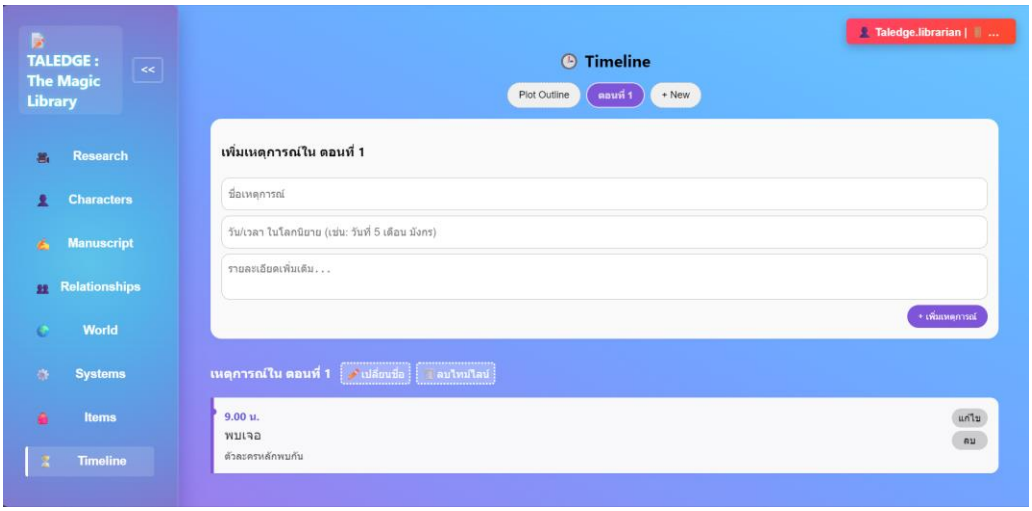
หน้าหลัก (Home)

เมื่อผู้ใช้เข้าสู่ระบบจะพบกับหน้าหลักพร้อมปุ่มสำหรับเพิ่มโปรเจกต์ใหม่หรือปุ่มแสดงรายการโปรเจกต์ทั้งหมดที่เคยสร้าง



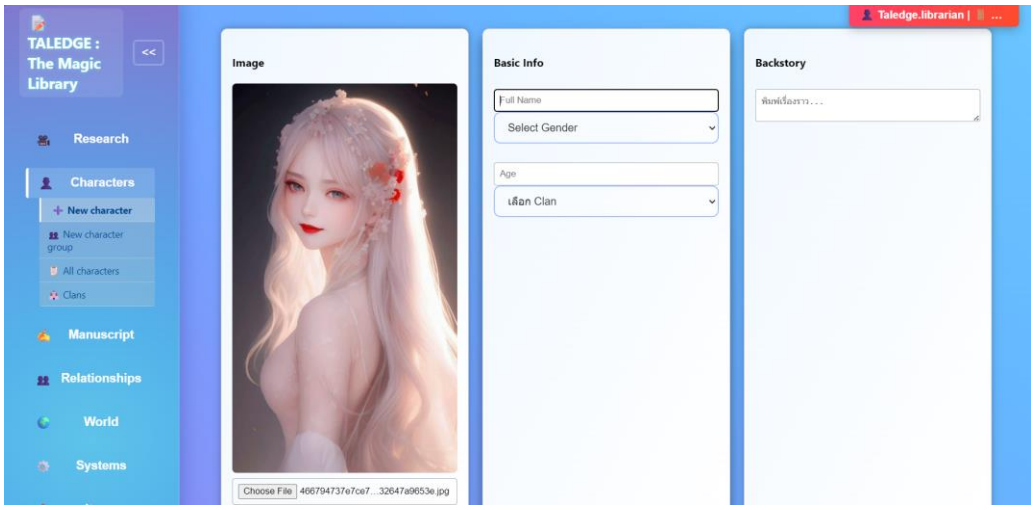
หน้าสร้างพล็อตเรื่องและจัดการเหตุการณ์ (Timeline)

ในหน้านี้ผู้ใช้สามารถเพิ่ม Timeline ของเรื่องราว แก้ไข หรือจัดเรียงเหตุการณ์ตามลำดับเวลาได้อย่างเป็นระบบ



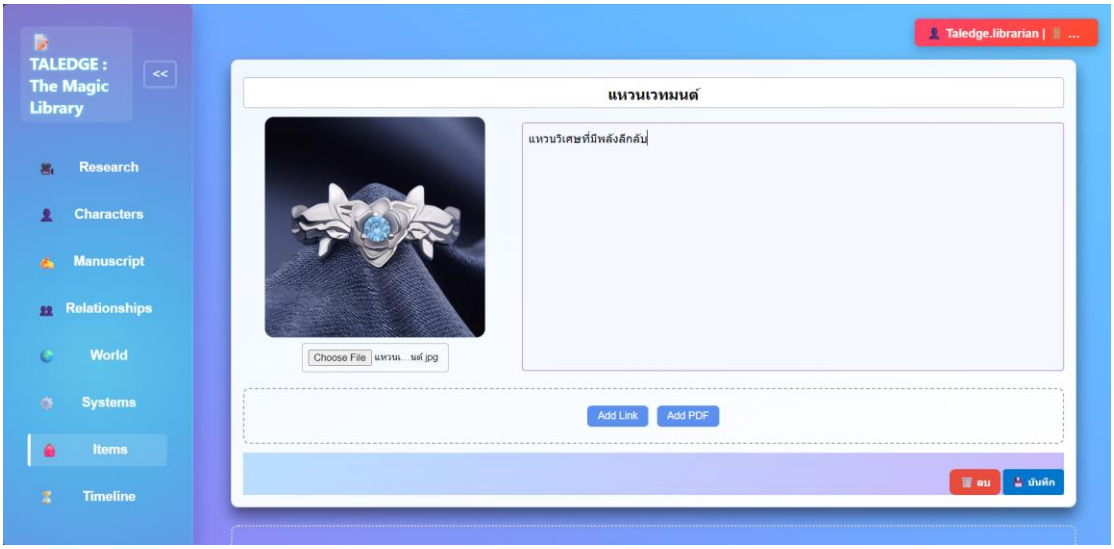
การจัดการตัวละคร (Characters Panel)

ผู้ใช้สามารถเพิ่มตัวละครใหม่ พร้อมกรอกข้อมูลสำคัญ เช่น ชื่อ อายุ ลักษณะภายนอก บุคลิก และความสัมพันธ์กับตัวละครอื่น ๆ ระบบมีการออกแบบให้รองรับการแสดงความเชื่อมโยงระหว่างตัวละครผ่าน Diagram แต่ในปัจจุบัน ระบบยังไม่สามารถแสดงความเชื่อมโยงระหว่างตัวละครแบบอัตโนมัติได้ ผู้ใช้ต้องเพิ่มความสัมพันธ์ด้วยตนเอง



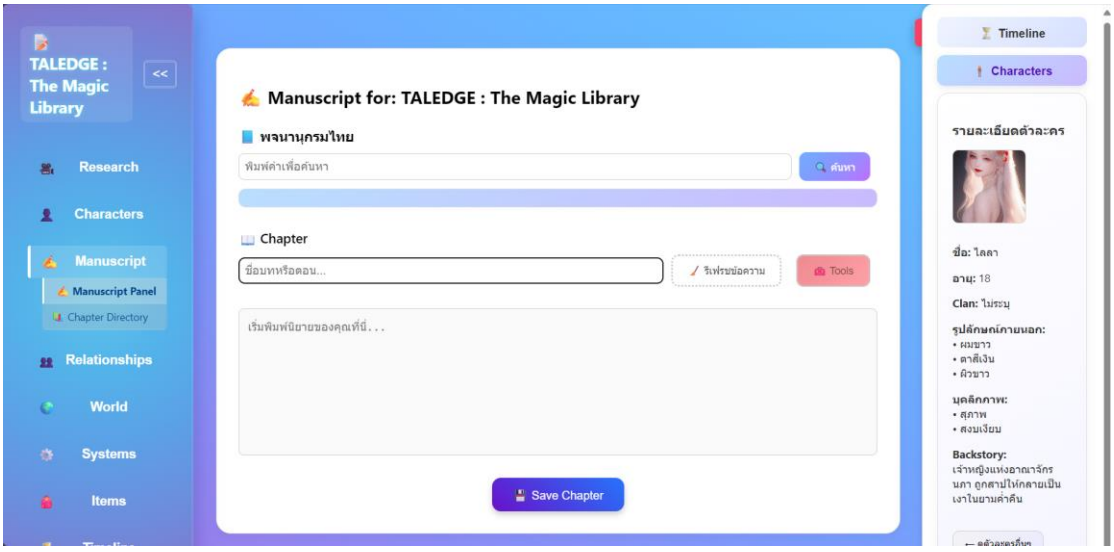
การจัดการไอเทมในเรื่อง (Items Panel)

ระบบรองรับการเพิ่มไอเทม เช่น อาวุธ สิ่งของวิเศษ เอกสาร ฯลฯ ผู้ใช้สามารถแนบรูปภาพ คำอธิบาย และลิงก์เอกสารหรือลิงก์วิดีโอเพิ่มเติมได้ในแต่ละไอเทม



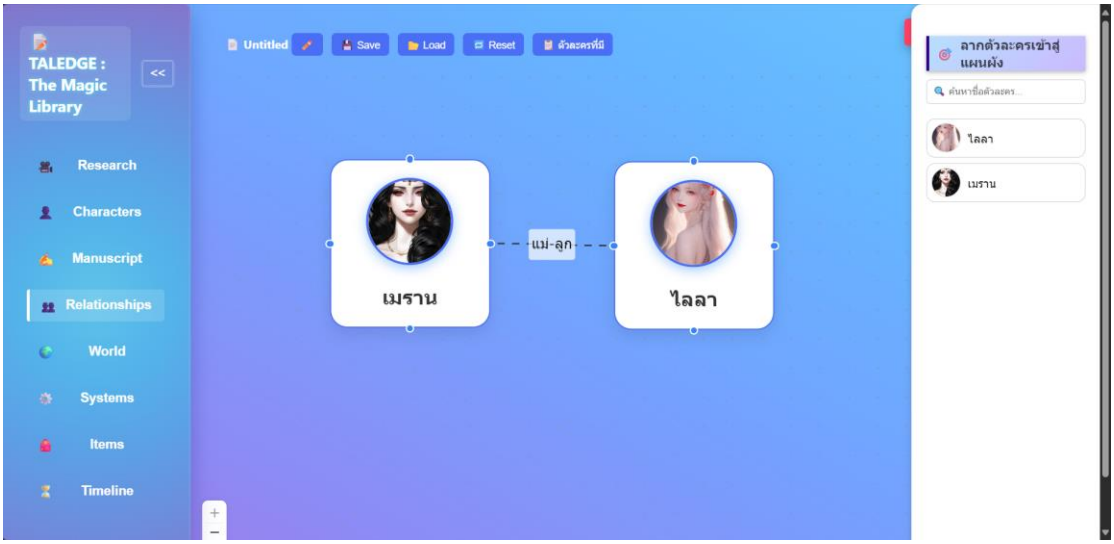
การเขียนต้นฉบับ (Manuscript Panel)

ผู้ใช้สามารถเริ่มต้นเขียนเนื้อหานิยายได้โดยตรงในระบบ โดยมี sidebar สำหรับแสดงข้อมูล Timeline และ Character ที่สร้างไว้มาช่วยอ้างอิงในการเขียน ระบบยังไม่มีฟีเจอร์ autosave ผู้ใช้ต้องกดบันทึกเองเมื่อเขียนเสร็จ



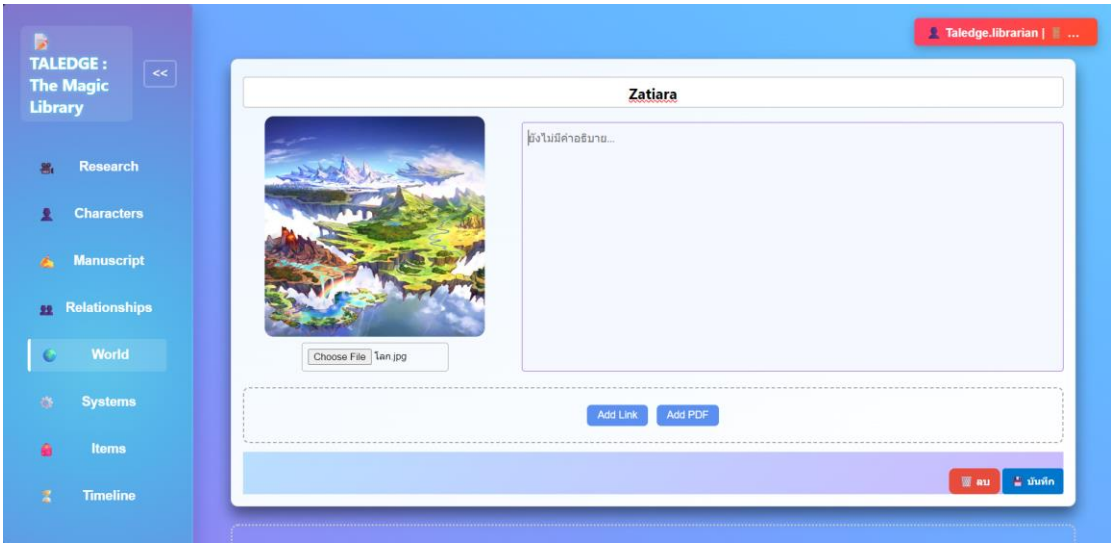
การเชื่อมโยงองค์ประกอบ

TALEDGE ได้พัฒนาให้สามารถเชื่อมโยงความสัมพันธ์รหว่างตัวละคร โดยมีการใช้เทคนิค drag-and-drop ตัวละครที่ได้จากตัวละครที่สร้างไว้มาใช้เพื่อความสะดวก



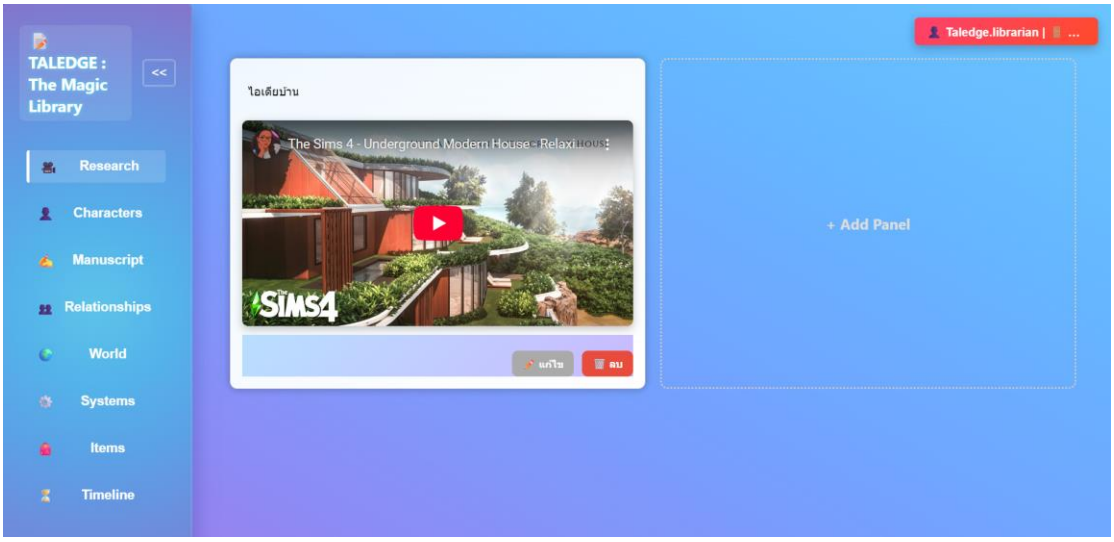
การจัดการโลกในเรื่อง (World Panel)

สร้างโลกนิยายของคุณเอง สร้างสถานที่ต่างๆ โดยผู้ใช้สามารถแนบรูปภาพ คำอธิบาย และลิงก์เอกสารหรือลิงก์วิดีโอเพิ่มเติมได้ในแต่ละการ์ดสถานที่



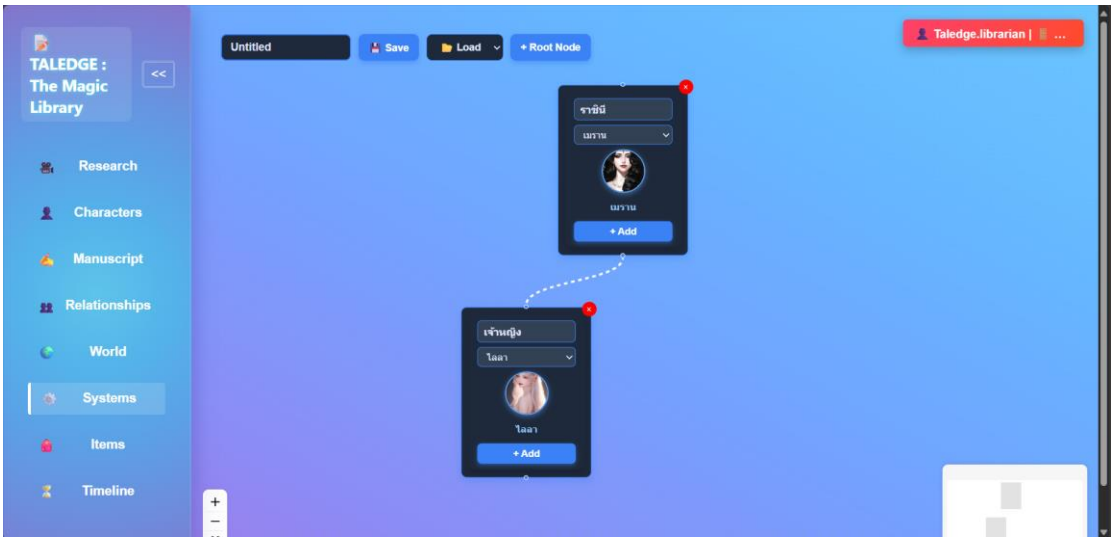
การจัดการไอเดีย (Research Panel)

ระบบ Research ออกแบบมาเพื่อให้ผู้ใช้สามารถเก็บข้อมูลอ้างอิงหรือไอเดียต่าง ๆ ได้อย่างเป็นระเบียบ เช่น การวางลิงก์, รูปภาพ, ไฟล์ PDF หรือบันทึกข้อความสั้น ๆ สำหรับใช้อ้างอิงในการแต่งนิยาย



ระบบ System (ผังระบบความสัมพันธ์หรือชนชั้น)

ระบบนี้ช่วยสร้างแผนผังของโครงสร้างสังคม เช่น ระบบชนชั้น องค์กร กลุ่มทางการเมือง หรือเผ่าพันธุ์ โดยใช้โครงสร้างแบบแผนภาพ เพื่อแสดงความสัมพันธ์แบบมีลำดับขั้น



v. ความท้าทาย อุปสรรค ที่พบในระหว่างพัฒนาระบบ

อุปสรรคที่พบในระหว่างพัฒนาระบบ

1. ความซับซ้อนในการเชื่อมโยงความสัมพันธ์ระหว่างตัวละคร
2. การจัดการ State ภายใน React เมื่อข้อมูลเปลี่ยนแปลงบ่อย
3. การออกแบบฐานข้อมูลให้รองรับความซับซ้อนของเนื้อหา
4. ความยากในการทำให้ UX รองรับนักเขียนที่มีสไตล์แตกต่างกัน
5. ปัญหาการเชื่อมต่อข้อมูลข้าม Component ของ React
6. การเก็บข้อมูลต้องแยกตาม projectID ด้วยจะแยกแค่ username ไม่ได้(ตอนแรกที่เก็บ schema คือ แยกแค่ username จะทำให้ระบบไม่รองรับการสร้างหลายๆ project เพราะจะทำให้ข้อมูลถูกเก็บรวมๆ ไว้ในที่เดียว
7. ในส่วน Fiction ของ Clan ไม่ขึ้นรูปภาพ เป็นเพราะไม่ได้เก็บแบบ ให้แปลงเป็นbase64 ทำให้รูปภาพไม่ขึ้นในส่วนนั้น เลยต้องแก้ในส่วนของ Handle saveImage
8. ในส่วนของ All Character ในตอนที่ Edit หลังจากEditแล้วกลายเป็นสร้างตัวละครใหม่ ไม่ได้ตรงตามวัตถุประสงค์ของการedit โดยต้องแก้ในส่วนของ routes backend
9. ในส่วนของ world กดตัว +Add panel แล้วพอเพิ่มข้อมูลแล้วเนี่ยไม่ขึ้นข้อมูลอะไรเลย เหมือนไม่ได้บันทึกเข้าในตัวของ schema ก็แก้ในส่วนของ schema พอดึงมาได้ ไม่ขึ้นรูปอีก ก็แก้ในส่วนของแปลงรูป base64 พอดึงรูปมาได้แล้ว ลบไม่ได้อีกต้องสร้าง คำสั่ง DELETE ในส่วนของ routes อีก
10. ในส่วนของ System ข้อมูลไม่ดึงจาก Database พอดึงแล้ว กดSave แต่รูปไม่ถูกเก็บใน mongo โหลดรูปตัวละครไม่ได้สร้างnodeไม่ได้ แก้ไข title ของแต่ละnode ไม่ได้ ดึงรูปของ child node ไม่ได้ ค่อนข้างยุ่งยากสุดๆ ต้องแก้ไขทั้งในส่วนของ Frontend ตั้งแต่การรับข้อมูล ไปจนถึง Backend คือการบันทึกไม่ว่าจะเป็น node edge หรือ title รูป

11. ส่วนของ Clan ไม่ดึงข้อมูลของ CharacterList คือไม่โชว์รูปตัวละครที่สร้าง จาก NewCharacter แต่แก้โดย ทำให้มัน fetch ข้อมูล จาก Character ด้วย ตอนแรกลืม
12. แก้ปัญหาโหลดข้อมูลไม่ขึ้น โดยแก้ในส่วนของ backend

ความท้าทายหลักๆ ที่พบในระหว่างพัฒนาระบบ

1. การทำในส่วนของ RelationshipEditor ซึ่งส่วนตัวคิดว่ายากสุดแล้วใน Project นี้ไม่ว่าจะเป็น Frontend หรือ Backend เพราะ Frontend ต้องมาเรียนรู้การใช้ React Flow ในการสร้างตารางกราฟแสดงความสัมพันธ์ อีกทั้งยังต้องสร้างหน้าตาของ Node รวมถึงการสร้างการเชื่อม EDGES **ซึ่งยากมาก** เพราะต้องเขียนโค้ดที่ทำให้แต่ละจุดเชื่อมกันได้ ไม่ว่าจะเป็นแบบ บน-ล่าง,ล่าง-บน,บน-บน อื่นๆ จนครบทุกจุด **ไม่พอ** ต้องเขียนโค้ดให้มันแยกด้วยว่าเป็นตัวเริ่มลาก หรือเป็นตัวลากไปหา ใช้เวลานานมากกว่าจะแก้ได้ และการใช้ node, edge เป็นความรู้ใหม่ทั้งหมดเพราะไม่เคยทำมาก่อนเลย
2. การเชื่อมตัว API Dictionary ไม่เคยทำมาก่อน ไม่ทราบเลยว่าสามารถทำได้ซึ่งอาจไม่ยากมาก เพราะในตัวของ python มีให้ใช้งานได้ค่อนข้างง่าย แต่ปัญหาคือ ไม่รองรับภาษาไทย และแก้ไม่ได้ พยายามที่จะเปลี่ยนให้เป็นภาษาไทยแล้วแต่ไม่สามารถทำได้จริงๆ
3. การเก็บข้อมูลแยก ในตอนแรกคิดว่าแค่แยกตาม username คือดีแล้วแต่ก็ต้องมีการแยกอีกเพราะไม่ครอบคลุมทั้งหมดทำให้ user ใช้งานได้ไม่สะดวก
4. การเขียน Login/Register/ForgotPassword ไม่เคยทำมาก่อน ในส่วนของระบบ ยิง OTP เป็นครั้งแรกที่ได้ลองใช้ component นี้ถือว่าค่อนข้างน่าสนใจสุดๆ

vi. สรุป

โครงงาน TALEDGE: ชั้นวางเรื่องราว ได้พัฒนาเว็บแอปพลิเคชันที่สามารถช่วยนักเขียนในการวางพล็อตและจัดการองค์ประกอบต่าง ๆ ของนิยายได้อย่างเป็นระบบ ตัวระบบรองรับการใช้งานได้ดีทั้งบนคอมพิวเตอร์ มีความพร้อมในการพัฒนาและขยายฟังก์ชันเพิ่มเติมในอนาคต เช่น ระบบทำงานร่วมกัน (Co-writing) หรือ AI Writing Assistant

อย่างไรก็ตาม ในขณะนี้ระบบยังได้ผ่านการทดสอบเฉพาะบนอุปกรณ์คอมพิวเตอร์ เท่านั้น และยังไม่ได้มีการนำไปทดสอบกับกลุ่มเป้าหมายจริง จึงยังไม่สามารถสรุปผลการ ใช้งานในบริบทของผู้ใช้ปลายทางได้อย่างครบถ้วน ซึ่งเป็นข้อจำกัดที่ควรพิจารณาเพิ่มเติม ในการพัฒนาและประเมินผลในอนาคต

ข้อเสนอแนะและแนวทางการพัฒนาต่อไป

1. เพิ่มระบบ "Co-writing" เพื่อให้ผู้ใช้งานสามารถทำงานร่วมกันในโปรเจกต์เดียวได้
2. พัฒนา AI Assistant สำหรับช่วยเสนอชื่อพล็อต หรือช่วยเขียนต่อ
3. ปรับปรุง UI ให้สามารถปรับธีม (Dark/Light Mode)
4. รองรับการ export ไฟล์เป็นรูปแบบ PDF หรือ ePub สำหรับการนำไปใช้งานต่อ
5. เพิ่มระบบ autosave เพื่อให้เนื้อหาถูกบันทึกอัตโนมัติ
6. เพิ่มพจนานุกรมภาษาไทย เพื่อช่วยนักเขียนในการค้นหาคำศัพท์และตรวจการสะกด

vii. ตารางสรุปความมีส่วนร่วมของนักศึกษา

ชื่อ-นามสกุล	หน้าที่และผลงานที่รับผิดชอบ
นายสิรวิชญ์ ทิมสุวรรณ	1. ออกแบบฐานข้อมูล 2. เขียน Frontend- Backend ในทุกระบบยกเว้น Timeline, Research, World 3. สร้าง REST API 4. เขียน Python API (Flask) 5. พัฒนาในส่วนของ Function Characters Panel, Manuscript Panel, ระบบความสัมพันธ์ตัวละคร(Relationships), System 6. แก้ไขปัญหาที่เกิดขึ้นจากการทดสอบระบบของฝ่ายทดสอบ 7. เขียนรายงานส่วนเทคนิค
นางสาวณิชาภัทร ชมภูน้อย	1. ออกแบบ UX/UI และ Mockup 2. พัฒนา CSS ทั้งระบบ 3. พัฒนา Frontend และ Backend ของ Timeline และ Research 4. พัฒนา Frontend ของ Items และ World 5. พัฒนาระบบลืมรหัสผ่าน 6. แก้ไขปัญหาที่เกิดขึ้นจาก Backend บางส่วนที่เพื่อนมองข้ามไป 7. เขียนรายงานส่วนออกแบบและทดสอบ